AD-A039 840 FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL AND SYS--ETC F/G 12/2
UNNETWORKS, WITH APPLICATIONS TO IDLE TIME SCHEDULING.(U)
APR 77 J J BARTHOLDI, H D RATLIFF N00014-76-C-0096

UNCLASSIFIED RR-77-4 NL

1 OF 1
ADA039 840

END
DATE
FILMED
6-77

RESEARCH REPORT

Industrial & Systems

(3)

UNNETWORKS, WITH APPLICATIONS TO IDLE TIME SCHEDULING

Research Report No. 77-4

by

John J. Bartholdi III
H. Donald Ratliff

April, 1977

Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

This research was supported in part by the Office of Naval Research under contract number N00014-76-C-0096 and the Army Research Office under contract number DAHC04-75-G-0150.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9 Research rept.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 77-4	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 14 RR-77-4
4. TITLE (and Subtitle) 6 Unnetworks, with Applications to Idle Time Scheduling		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER 77-4
7. AUTHOR(s) 10 John J. Bartholdi III, H. Donald Ratliff		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-76-C-0096, DAHC04-75-G-0150
9. PERFORMING ORGANIZATION NAME AND ADDRESS Industrial & Systems Engineering, University of Florida, Gainesville, Florida 32611		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 20061102A14D ~~Rsch in & Appl of Applied Math.~~
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office, P. O. Box 12211, Triangle Park, NC 27709; Office of Naval Research, Arlington, VA		12. REPORT DATE 11 April 1977
		13. NUMBER OF PAGES 12 21
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

N/A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

integer programming
complementary problem
network flows
idle time scheduling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Dormouse celebrated its unbirthday 364 days of the year. Accordingly, this class of integer linear programs might be called unnetworks. They are interesting and efficiently solvable because of what is not there.

Applications to idle time scheduling are discussed.

404399

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Table of Contents

Abstract

The Dormouse celebrated its unbirthday 364 days of the year. Accordingly, this class of integer linear programs might be called <u>un</u>networks. They are interesting and efficiently solvable because of what is <u>not</u> there.

Applications to idle time scheduling are discussed.

0. Introduction

In solving an integer linear program with a 0-1 constraint matrix, sufficient conditions for tractability seem to emphasize the pattern of 1's within the matrix (e.g. Hoffman and Kruskal [10], Iri [11]). But by the same token, the pattern of 0's may predispose a problem to tractability. In general this has been overlooked, perhaps because of the natural tendency to concentrate on "what is there" rather than on what is not. We make a simple change of focus to define a complementary problem; by so doing, we identify a previously unrecognized class of efficiently solvable integer linear programs.

1. Motivation

Recently a manpower scheduling problem studied by Tibrewala, Philippe, and Browne [13] has generated considerable discussion in the literature (Baker [2], Brownell and Lowerre [4], Chen [5]). The problem is to minimize the number of workers needed to meet daily requirements with the added proviso that each worker must be allowed two consecutive days off each week. This may be modelled as an integer linear program:

$$\min \ \bar{1}\bar{x}$$

$$\text{s.t.} \quad \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \bar{x} \geq \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \qquad (1.1)$$

$$\bar{x} \geq \bar{0}, \quad \text{integer}$$

where each column represents a possible pattern of days off, x_j represents the number of workers on work pattern j, and b_i is the number of workers required on day i.

Now while Tibrewala, et al. have in fact developed a simple numerical algorithm for solving this problem, the integer linear program formulation is, on first glance somewhat distressful: the constraint matrix displays none of those immediately commendable virtues for which one might hope, i.e., it is not unimodular, balanced, or even perfect (Padberg [12]). Nevertheless, its special structure is appealing. We observe that there are two zeros in each column. Recalling that a network may be uniquely associated with a matrix having two ones in each column, we may imagine that our matrix determines everything but a network - an unnetwork, if you will.

Consider now the complementary problem, which is a sort of negative image of the original:

$$\max \ \bar{1}\bar{y}$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \bar{y} \leq \begin{bmatrix} s - b_1 \\ s - b_2 \\ s - b_3 \\ s - b_4 \\ s - b_5 \\ s - b_6 \\ s - b_7 \end{bmatrix} \quad (1.\ 2)$$

$$\bar{y} \geq \bar{0}, \text{ integer}$$

Imagine s to be a workforce size (positive integer, of course); then $s - b_i$ represents the maximum number of workers who may be idle on day i. We then have formulated the following problem: given a workforce of size s, maximize the number of workers who may be given two consecutive days off each week. Clearly, if for given s, the solution satisfies $\bar{1}\bar{y} < s$, the workforce is too small to meet daily staffing requirements and still allow everyone time off. On the other hand, if for fixed s we find $\bar{1}\bar{y} \geq s$, we may conclude that the workforce is adequate. If in fact $\bar{1}\bar{y} > s$, we may interpret this to say that too many workers have been assigned time off. A feasible time-off schedule for a work force of size s may be easily found, however, by simply reducing the entries in $\bar{y}$ to $\bar{y}_r$ such that $\bar{1}\bar{y}_r = s$; that is, time-off is assigned to no more than s workers. What we seek then is the smallest integer $s = s^*$ for which the corresponding solution to (1.1) satisfies $\bar{1}\bar{y}^* \geq s^*$, i.e., the smallest adequate work force. We are encouraged to search for this s* by the happy fact that the derived problem is a familiar one, viz. a maximum matching problem on a graph (Edmonds [6]).

Let us take a more general and formal view of the idea suggested here.

2. A Primal Problem and Its Complement

Let A be a 0-1 matrix of m rows and n columns, and let $\bar{b}$ be a compatibly dimensioned vector with all entries integer. Consider the integer linear program:

$$\begin{aligned} & \min \bar{1}\bar{x} \\ \text{s.t.} \quad & A\bar{x} \geq \bar{b} \qquad (2.1) \\ & \bar{x} \geq \bar{0}, \text{ integer} \end{aligned}$$

where, without loss of generality, we may assume $\bar{b} > \bar{0}$.

Let E be the m x n matrix whose every entry is 1, and let $\bar{1}$ and $\bar{s}$ be compatibly dimensioned vectors whose every entry is 1 and s, respectively. Then the following integer linear program - which we call the complement of (2.1) - is of interest.

$$\begin{aligned} & \max \bar{1}\bar{y} \\ \text{s.t.} \quad & [E-A]\bar{y} \leq \bar{s}-\bar{b} \qquad (2.2) \\ & \bar{y} \geq \bar{0}, \text{ integer} \end{aligned}$$

Note that the matrix $E-A = A^c$ (read "A complement") has 1's wherever A has 0's and has 0's wherever A has 1's.

The following results then pertain:

Theorem 2.1: Let s^* be the smallest integer s for which the corresponding solution $\bar{y}^*$ to (1.2) satisfies $\bar{1}\bar{y}^* \geq s^*$. Reduce the entries in $\bar{y}^*$ arbitrarily - though maintaining integrality and non-negativity - to $\bar{y}^*_r$ such that $\bar{1}\bar{y}^*_r = s^*$. Then $\bar{x}^* = \bar{y}^*_r$ solves problem (1.1) and the optimal objective function value is $\bar{1}\bar{x}^* = \bar{1}\bar{y}^*_r = s^*$.

Proof: By construction $\bar{y}^*_r \geq \bar{0}$ and integer; and since $\bar{0} \leq \bar{y}^*_r \leq \bar{y}^*$, we have that $[E-A]\bar{y}^*_r \leq \bar{s}^*-b$. This together with $\bar{1}\bar{y}^*_r = s^*$ iff $E\bar{y}^*_r = \bar{s}^*$, implies that $A\bar{y}^*_r \geq \bar{b}$, so that $\bar{y}^*_r$ is feasible to problem (2.1). Now if $\bar{y}^*_r$ is not optimal to (2.1), there must exist some $\bar{y}$ feasible to (2.1) such that $\bar{1}\bar{y} < \bar{1}\bar{y}^*_r = s^*$. But then $\bar{y}$ is feasible to problem (2.2) for $s = s' = \bar{1}\bar{y}$, so that the optimal solution $\bar{y}'$ to (2.2) for $s = s'$ satisfies $\bar{1}\bar{y}' \geq \bar{1}\bar{y} = s'$. But this contradicts the assumption that s^* is the smallest integer value of s for which such a solution exists.

Q.E.D.

Therefore, if we can locate the smallest $s = s^*$ for which the corresponding solution $\bar{y}^*$ to problem (2.2) satisfies $\bar{1}\bar{y}^* \geq s^*$, the solution to the original problem is at hand. In addition, when $\bar{1}\bar{y}^* > s^*$, lemma 2.1 implies that we have great freedom in moving to an optimal solution $\bar{y}^*_r$. Hence, we have a host of easily attainable alternative optima from which we may choose in order to satisfy secondary criteria.

A famous recipe for hare stew begins, "First you catch a hare..." In this case we must locate s^*. Fortunately an efficient trap is provided by the several results to follow.

Let b_{max} denote the largest entry in $\bar{b}$. Then

Lemma 2.1: $b_{max} \leq s^* \leq \min(\bar{1}\bar{b}, nb_{max})$

Proof: To establish the upper bound, notice that at optimality every variable in problem (2.1) appears in some tight constraint, since otherwise that variable could be reduced and feasibility maintained. Summing the set T of tight constraints yields $\sum_{i \in T} \sum_j a_{ij} x_j^* = \sum_{i \in T} b_i$; but $\bar{1}\bar{b} \geq \sum_{i \in T} b_i = \sum_{i \in T} \sum_j a_{ij} x_j^* \geq \sum_j x_j^* = \bar{1}\bar{x} = s^*$. Also, letting $b_i = \sum_j a_{ij} x_j^*$ be a tight constraint in which $\bar{x}_j^*$ appears, we have that $b_i = \sum_j a_{ij} x_j^* \geq x_j^*$ so that certainly $b_{max} \geq x_j^*$ and $nb_{max} \geq \bar{1}\bar{x}^* = s^*$.

To show the lower bound, it is sufficient to observe that since $\bar{x} \geq \bar{0}$, $E\bar{x} \geq A\bar{x} \geq \bar{b}$ so that $\bar{1}\bar{x} \geq b_j \ \forall \ j$. Thus $s^* = \bar{1}\bar{x}^* \geq b_{max}$

Q.E.D.

Incidentally, these bounds are <u>in general</u> the best possible, since if $A = E$, then $s^* = b_{max}$ while if $A = I$, $s^* = \bar{1}\bar{b}$. Of course for specially structured matrices A, tighter bounds may be derived. At any rate, we may conclude that both the range and the magnitude of the integer s^* are bounded by a polynomial in the values of the problem data.

Lemma 2.2: For fixed integer s', let $\bar{y}'$ be the corresponding solution to problem (1.2). Then $\bar{1}\bar{y}' < s'$ iff $s' < s^*$, where s^* is defined in Theorem 2.1.

Proof: If $s' < s^*$ then $\bar{1}\bar{y}' < s'$ since s^* is the smallest integer less than or equal to its corresponding optimal objective function value $\bar{1}\bar{y}^*$.

On the other hand, suppose $\bar{1}\bar{y}' < s'$ but $s^* \leq s'$. Then $\bar{y} = \bar{y}^* + \begin{bmatrix} s'-s^* \\ \bar{0} \end{bmatrix}$

satisfies $\bar{y} \geq \bar{0}$, and $[E-A]\bar{y} = [E-A]\bar{y}^* + [E-A] \begin{bmatrix} s'-s^* \\ \bar{0} \end{bmatrix} \leq s^* - \bar{b} + E \begin{bmatrix} s'-s^* \\ \bar{0} \end{bmatrix} =$

$\bar{s}^* - \bar{b} + \bar{s}' - \bar{s}^* = \bar{s}' - \bar{b}$; thus $\bar{y}$ is feasible to problem (2.2) with $s = s'$. Moreover $\bar{1}\bar{y} = \bar{1}\bar{y}^* + s' - s^* \geq s^* + s' - s^* \geq s' > \bar{1}\bar{y}'$, which contradicts $\bar{y}'$ solving (2.2) for $s = s'$.

Q.E.D.

Corollary: For fixed integer s', let $\bar{y}'$ be the corresponding solution to (2.2). Then $\bar{1}\bar{y}' \geq s'$ iff $s^* \leq s'$.

Thus the relationship between s and its associated optimal objective function value for problem (2.2) may be illustrated as in Figure 2.1.

3. An Algorithm Based on the Complementary Problem

We incorporate the preceding results in

A Complementary Algorithm

Step 0: Given the problem

$$\begin{aligned} &\min \bar{1}\bar{x} \\ &\text{s.t. } A\bar{x} \geq \bar{b} \\ &\qquad \bar{x} \geq \bar{0}, \text{ integer} \end{aligned} \tag{2.1}$$

form the complementary problem

$$\begin{aligned} &\max 1\bar{y} \\ &\text{s.t. } [E\text{-}A]\bar{y} \leq \bar{s} - \bar{b} \\ &\qquad \bar{y} \geq \bar{0}, \text{ integer} \end{aligned} \tag{2.2}$$

Step 1: Search for s*

A. Restrict s* to the interval $b_{max} \leq s^* \leq \min(\bar{1}\bar{b}, nb_{max})$ where s* is an integer.

B. Perform binary search (Aho, Hopcroft, and Ullman [1]) through this interval to locate s*. At each iteration s is fixed at a value s' and the corresponding version of problem (2.2) is solved; then the optimal objective function value $\bar{1}\bar{y}'$ is compared to s' and lemma 2.2 invoked to further restrict the location of s* to $s^* \leq s'$ or $s' < s^*$.

Having determined s* and the corresponding solution $\bar{y}^*$ to problem (2.2), proceed to step 2.

Step 2: Construct the optimal solution to the original problem, (2.1).

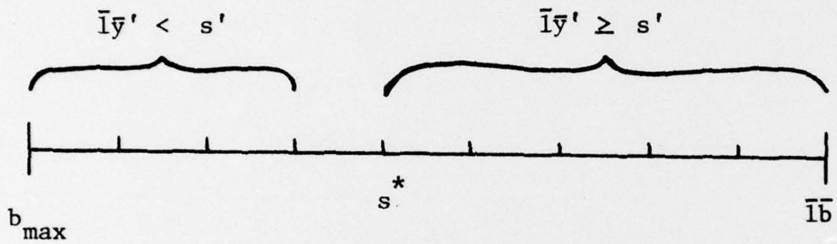

Figure 2.1: Behavior of s' and associated optimal solution $\bar{y}'$ (see lemma 2.2 and corollary).

A. By theorem 2.1, we may reduce entries in y^* arbitrarily - though maintaining integrality and non-negativity - to construct $\bar{y}^*_r$ such that $\bar{1}\bar{y}^*_r = s^*$. Then $\bar{x}^* = \bar{y}^*_r$ is the solution to the original problem, (2.1).

4. Efficiency of the Algorithm

An algorithm may be considered formally efficient if its worst case performance may be bounded above by a polynomial in the size of an encoding of the problem data (Aho, Hopcroft, and Ullman [1]). Accordingly, we may consider an algorithm binary efficient if it is efficient with respect to a binary encoding; similarly, unary efficient if efficient with respect to a unary encoding. The minimum cost network flow algorithm for instance is binary efficient (Edmonds and Karp [8]) and the maximum b-matching is unary efficient (Edmonds and Johnson [7]).

Lemma 4.1: If, for any fixed s, the complementary problem, (2.2), is efficiently solvable with respect to a binary encoding of the problem data, then the complementary algorithm solves the primal problem, (2.1), with formal efficiency relative to a binary encoding.

Proof: Since assumed efficiently solvable with respect to a binary encoding, suppose the complementary problem (2.2) is solvable in no more than $O(p(n, \log_2 \bar{1}\bar{b}, \log_2 s))$ steps for some polynomial p, which we may assume to be non-decreasing in its variables, n, $\log_2 \bar{1}\bar{b}$, $\log_2 s$. Since by lemma 2.1, $s^* \leq \bar{1}\bar{b}$, it must be that $\log_2 s^* \leq \log_2 \bar{1}\bar{b}$. Thus each complementary problem solved by the algorithm requires no more than $O(p(n, \log_2 \bar{1}\bar{b}, \log_2 \bar{1}\bar{b}))$ steps--or simply $O(\hat{p}(n, \log_2 \bar{1}\bar{b}))$ for some polynomial $\hat{p}$.

Now, to perform binary search for s^* in the interval $b_{max} \leq s^* \leq \bar{1}\bar{b}$, we need solve no more than $O(\log_2 \bar{1}\bar{b})$ complementary problems; thus, no more than $O(\log_2 \bar{1}\bar{b} \cdot \hat{p}(n, \log_2 \bar{1}\bar{b}))$ steps are required to find s^* and the accompanying solution to the appropriate complementary problem.

Finally, constructing $\bar{y}_r^*$ as described in theorem 2.1 is a simple $O(n)$ process. Therefore, no more than $O(\log_2 \bar{1}\bar{b} \cdot \hat{p}(n, \log_2 \bar{1}\bar{b}) + n)$ steps are required by the complementary algorithm to solve the primal problem. Clearly this is polynomial in terms of a binary encoding of the data of the primal problem.

Q.E.D.

Corollary: If, for any fixed s, the complementary problem (2.2) is efficiently solvable with respect to a unary encoding of the problem data, then the complementary algorithm solves the primal problem, (2.1), with formal efficiency relative to a unary encoding.

To efficiently solve the primal then, it is enough that there exist some efficient solution technique for the complement.

5. A Round-Off Result

A rather surprising round-off result, similar to that obtained by Weinberger [15], holds for special versions of problem (2.1).

Consider the continuous-valued relaxations of problems (2.1) and (2.2):

$$\begin{aligned} \min \quad & \bar{1}\bar{x} \\ \text{s.t.} \quad & A\bar{x} \geq \bar{b} \\ & \bar{x} \geq \bar{0} \end{aligned} \tag{5.1}$$

and

$$\begin{aligned} \max \quad & \bar{1}\bar{y} \\ \text{s.t.} \quad & [E-A]\bar{y} \leq \bar{s} - \bar{b} \\ & \bar{y} \geq \bar{0} \end{aligned} \tag{5.2}$$

respectively. Then letting z^* be the optimal function value of the continuous-valued problem, (5.1) and s^* that of the integer-restricted version, (2.1),

Theorem 5.1: If for any integral s, problem (5.2) has an optimal point that is integral, then $\lceil z^* \rceil = s^*$, where $\lceil \cdot \rceil$ is rounded up to the closest integer.

Proof: Let $\bar{x}^*$ solve (5.1) with $\bar{1}\bar{x}^* = z^*$. Then by an argument similar to that of lemma 2.1, $\bar{y}' = \bar{x}^* + \begin{bmatrix} \lceil z \rceil - z^* \\ \bar{0} \end{bmatrix}$ is feasible to (5.2) for $s = \lceil z^* \rceil$ and

$\overline{1y}' = \lceil z^* \rceil$. Thus the optimal solution to (5.2) for $s = \lceil z^* \rceil$ must satisfy $\overline{1y} \geq \lceil z^* \rceil$. Moreover, by the assumed property of (5.2), we may take this solution to be integer and therefore feasible to (2.2). But then by the corollary to lemma 2.2, we have that $s^* \leq \lceil z^* \rceil$. But since (5.1) is a relaxation of (2.1), $z^* \leq s^*$. Therefore we conclude that $s^* = \lceil z^* \rceil$.

Q.E.D.

For matrices such that E-A is totally unimodular, (5.2) has integral extreme points and the round-off result applies. A particular class of such matrices are those for which the complementary problems are solvable by the minimum cost network flow algorithm. Iri [11] has given a characterization of such matrices along with a proof of their total unimodularity. Example 6b in the following section illustrates this property.

6. Examples

(a) To return to the two consecutive days off scheduling problem, (1.1): it is clear that this may be solved by the complementary algorithm as a bounded series of maximum $\bar{b}$-matchings (or, in this case, maximum $\bar{s} - \bar{b}$ matchings). It may be further observed that the complement, (1.2), poses the matching problem on a special graph, *viz.* a seven node simple cycle. Moreover it is not difficult to see that the algorithms of Tibrewala, *et al.* and Chen are essentially maximum matching algorithms for a simple cycle with node constraints $\bar{s} - \bar{b}$. Thus their algorithms may be considered a special implementation of the complementary algorithm where the search for s^* is speeded up by special structure.

For the cases in which $\overline{1y}^* > s^*$, the complementary algorithm offers us, in addition to alternative solutions, some special insight into the staffing problem. We may interpret the condition of $\overline{1y}^*$ being much greater than s^* to indicate an incompatibility between the work patterns and the pattern of manpower requirements. For such cases, it may be advisable for the manager to consider differently structured work patterns, or else reconsider the data in $\bar{b}$.

Incidentally, it is clear that the complementary algorithm will also efficiently solve this problem even when two non-consecutive days off are allowed. In this case, the complement simply poses a matching problem on a graph more general than a simple cycle.

(b) Consider a set of machines that are to be scheduled over a finite horizon to meet machine requirements $\bar{b}$. Each machine has peculiar to itself some number of consecutive time periods during which it must remain idle - for preventive maintenance for instance. To find the minimum number of machines necessary, we may model the problem as in this example:

$$\begin{aligned} &\min \bar{1}\bar{x} \\ \text{s.t.} &\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \bar{x} \geq \bar{b} \\ &x_j = 0 \text{ or } 1 \end{aligned} \tag{6.1}$$

where column j reflects the availability of machine j over the discretized planning horizon, and where, in any solution, $x_j = 1$ corresponds to utilizing machine j. The key feature of this example is that A possesses the property of consecutive 0's in columns.

For (6.1) the complement is

$$\max \bar{1}\bar{x}$$

$$\text{s.t.} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \bar{x} \leq \bar{s} - \bar{b} \qquad (6.2)$$

$$x_j = 0 \text{ or } 1$$

for which the constraint matrix possesses the well-known "consecutive 1's" property (Fulkerson and Gross [9]). Thus (6.2) is transformable to a minimum cost network flow problem as in Veinott and Wagner [14]. We conclude then that (6.1) is efficiently solvable as a bounded series of minimum cost network flow problems, and, moreover, that the round-off result of section 5 holds.

7. Another Staffing Problem

To indicate the flexibility of the preceeding approach, consider a problem studied by Brownell and Lowerre [4]. They seek to minimize the number of workers required to meet daily staffing requirements where each workers is allotted two days off each week including every other weekend. They solve the special case in which all weekday staffing requirements are identical and weekend requirements are identical. However, for the more general case of arbitrary daily requirements, a variation of the complementary algorithm provides an efficient solution.

The problem may be posed as an integer program with 0 - 1 constraint matrix thusly:

$$\min \bar{1}\bar{x}_1 + \bar{1}\bar{x}_2$$

$$\begin{array}{r} \text{s.t. (1st week)} \\ \text{(1st weekend)} \\ \text{(2nd week)} \\ \text{(2nd weekend)} \end{array} \quad \left[\begin{array}{c|c} E & \\ \cline{1-1} 0 & A_2 \\ \hline & E \\ \cline{2-2} A_1 & 0 \end{array}\right] \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} \geq \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \\ \bar{b}_4 \end{bmatrix} \qquad (7,1)$$

$$\bar{x}_1, \bar{x}_2 \geq \bar{0}, \quad \text{integer}$$

where each column represents a possible pattern of days off, $\bar{x}_1$ corresponds to shifts with the first weekend off, and $\bar{x}_2$ corresponds to shifts with the second weekend off. Furthermore, A_1 and A_2 are 0-1 matrices with exactly two zeros in each column, corresponding to two days off for the week.

Now let s_1 be the number of workers on shift 1 (first weekend off) and let s_2 be the number of workers on shift 2, so that $s = s_1 + s_2$ is the size of the total workforce. We observe that during the weekdays of the first week both shift 1 and shift 2 workers are employed; thus to meet staffing requirements, no more than $\bar{s} - \bar{b}_1 = \bar{s}_1 + \bar{s}_2 - \bar{b}$, workers (entry-wise) may be given time off on these days. On the days of first weekend, only shift 2 workers are on duty, so that no more than $\bar{s}_2 - \bar{b}_2$ workers (entry-wise) may be off those days. Similarly, $\bar{s}_1 + \bar{s}_2 - \bar{b}_3$ and $\bar{s}_1 - \bar{b}_4$ give upper bounds on the number of workers who may be off during the days of the second week.

As in section 0, we may ask: for a given workforce of size s, partitioned into two shifts s_1 and s_2 ($s = s_1 + s_2$), what is the maximum number of people who may be given time off according to the work patterns and requirements above? Again, analagously to section 0, this may be posed mathematically as

$$\max\ \bar{1}\bar{y}_1 + \bar{1}\bar{y}_2$$

$$\text{s.t.} \quad \left[\begin{array}{c|c} 0 & E-A_2 \\ \hline 0 & \\ \hline E-A_1 & 0 \\ \cline{2-2} & 0 \end{array}\right] \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} \leq \begin{bmatrix} \bar{s}_1 + \bar{s}_2 - \bar{b}_1 \\ \bar{s}_2 - \bar{b}_2 \\ \bar{s}_1 + \bar{s}_2 - \bar{b}_3 \\ \bar{s}_1 \quad\quad - \bar{b}_4 \end{bmatrix} \tag{7.2}$$

$$\bar{y}_1, \bar{y}_2 \geq \bar{0}, \text{ integer}$$

which may be decoupled into the two problems

$$\max\ \bar{1}\bar{y}_1$$

$$\text{s.t.} \quad [E - A_1]\bar{y}_1 \leq \begin{bmatrix} \bar{s}_1 + \bar{s}_2 - \bar{b}_3 \\ \bar{s}_1 \quad\quad - \bar{b}_4 \end{bmatrix} \tag{7.3a}$$

$$\bar{y}_1 \geq \bar{0}, \text{ integer}$$

and

$$\max\ \bar{1}\bar{y}_2$$

$$\text{s.t.} \quad [E - A_2]\bar{y}_2 \leq \begin{bmatrix} \bar{s}_1 + \bar{s}_2 - \bar{b}_1 \\ \bar{s}_2 - \bar{b}_2 \end{bmatrix} \tag{7.3b}$$

$$\bar{y}_2 \geq \bar{0}, \text{ integer}$$

Problems (7.3a and b) are similar to that discussed in example 6a. In this case an adequate workforce $s = s_1 + s_2$ must have corresponding solutions to (7.3a and b) which satisfy $\bar{1}\bar{y}_1 \geq s_1$ and $\bar{1}\bar{y}_2 \geq s_2$. Thus we must search for the smallest integer s together with a partition $s = s_1 + s_2$ for which these conditions are satisfied. Such an s and accompanying partition may be found by applying two-dimensional binary search in the following manner: for fixed s, s_1 is restricted to the interval $0 \leq s_1 \leq \lceil s/2 \rceil$; on this interval binary search is performed, solving at each step the appropriate versions of (7.3a and b); if am acceptable partition, $s = s_1 + s_2$, is found, then s is an adequate workforce and $s^* \leq s$; if no acceptable partition is found then s is not adequate and $s < s^*$. Finally, having determined $s^* = s_1^* + s_2^*$ and the corresponding solutions $\bar{y}_1^*$ and $\bar{y}_2^*$ to (7.3a and b), an optimal solution to (7.1) is constructed as in theorem 2.1. In a manner analagous to section 2, the preceeding may be formally argued.

If $O(\beta(n, \log_2 \bar{1}\bar{b}))$ reflects the complexity of problem (7.3), then as in section 5, it may be argued that this solves the Brownell and Lowerre problem in no more than $O((\log \bar{1}\bar{b})^2 \cdot \beta(n, \log_2 \bar{1}\bar{b}) + n)$ steps. Again this is formally efficient with respect to a unary encoding since the problems (7.3) are matching problems.

Additionally, it may be noted that this approach solves the "k - 1 out of k" weekend problem in no more than $O((\log_2 \bar{1}\bar{b})^k \cdot \beta(n, \log_2 \bar{1}\bar{b}) + n)$ steps.

8. Extensions and Applications

While the above analysis was carried out for (2.1) a minimization problem with "$\geq$" constraints, similar results hold for maximization problems with "$\leq$" constraints. In addition, several technical extensions of this approach are possible (Bartholdi [3]). For instance, in some cases one may choose to flip the 0's and 1's in just a portion of the constraint matrix, as was done in section 7. Also, for the case of arbitrarily weighted objective functions $\bar{c}\bar{x}$, a smaller but still useful class of problems may be solved by a similar idea.

The particular applicability of this technique seems to lie in scheduling/staffing problems, where it appears natural to model job/processor availability by 0 - 1 matrices. As illustrated here, many of the basic problem have appeallingly patterned matrices. This is especially true of the fundamental models of idle time scheduling -- the results of which are unified and extended by this complementary approach.

Finally, an area of obvious applicability would be to problems with dense

0 - 1 matrices. The complements would of course be sparse and therefore perhaps amenable to (at least empirically efficient) solution.

Bibliography

1. Aho, A. V., J. E. Hopcroft, J. D. Ullman: The Design and Analysis of Computer Algorithms, Addison-Wesley Publishing Company, 1974.

2. Baker, K. R.: "Scheduling a Full-Time Work Force to Meet Cyclic Staffing Requirements," Management Science, Vol. 20, No. 12, August, 1974, pp. 1561-8.

3. Bartholdi, J. J., III: Doctoral Dissertation, Department of Industrial and Systems Engineering, University of Florida, Gainesville, Florida, 1977.

4. Brownell, W. S. and J. M. Lowerre: "Scheduling of Workforce Required in Continuous Operations Under Alternative Labor Policies," Management Science, Vol. 22, No. 5, January, 1976, pp. 597-605.

5. Chen, D. S.: "A Simple Manpower Scheduling Algorithm for Two Consecutive Days Off," presented at the ORSA/TIMS Joint National Meeting, Miami, Florida, November, 1976.

6. Edmonds, J.: "Paths, Trees, and Flowers," Canadian Journal of Mathematics, Vol. 17, 1965, pp. 449-67.

7. Edmonds, J. and E. L. Johnson: "Matching: A well-Solved Class of Integer Linear Programs," in Combinatorial Structures and Their Applications, Gordon and Breach, New York, 1970, pp. 89-92.

8. Edmonds, J. and R. M. Karp: "Theoretical Improvements in Algorithmic Efficiency fo Network Flow Problems," Journal for the Association of Computing Machinery, Vol. 19, No. 2, April, 1972, pp. 248-64.

9. Fulkerson, D. R. and O. A. Gross: "Incidence Matrices and Interval Graphs," Pacific Journal of Mathematics, Vol. 15, No. 3, 1965, pp. 835-55.

10. Hoffman, A. J. and J. B. Kruskal: "Integral Boundary Points of Convex Polyhedra," in H. W. Kuhn and A. W. Tucker (eds.): Linear Inequalities and Related Systems, Annals of Mathematics Studies, No. 38, Princeton, New Jersey, 1956, pp. 223-46.

11. Iri, M: "A Criterion for the Reducibility of a Linear Programming Problem to a Linear Network Flow Problem," RAAG Research Notes, Third Series, No. 98, February, 1966.

12. Padberg, M. W.: "Characterizations of Totally Unimodular, Balanced and Perfect Matrices," in B. Roy (ed.): Combinatorial Programming, D. Reidel Publishing Company, pp. 275-84.

13. Tibrewala, R., D. Philippe, and J. Browne: "Optimal Scheduling of Two Consecutive Idle Periods," Management Science, Vol. 19, No. 1, September, 1972, pp. 71-5.

14. Veinott, A. F., Jr., and H. M. Wagner: "Optimal Capacity Scheduling - I and II," Operations Research, Vol. 10, No. 4, 1962, pp. 518-46.

15. Weinberger, D. B.: "Network Flows, Minimum Coverings, and the Four Color Conjecture," Operations Research, Vol. 24, No. 2, March - April, 1976, pp. 272-90.